Joint Coding for Proactive Caching with Changing File Popularities

Published in:
2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)

Publication Status:
Published (in print/issue): 15/02/2018

DOI:
10.1109/PIMRC.2017.8292548

Document Version
Author Accepted version

Joint Coding for Proactive Caching with Changing File Popularities

Mohamed Amir
Faculty of Engineering
Memorial University,
St. Johns, NL, Canada.
makhalil@mun.ca

Ebrahim Bedeer
School of Engineering,
Ulster University,
United Kingdom.
e.bedeer.mohamed@ulster.ac.uk

M. H. Ahmed
Faculty of Engineering
Memorial University,
St. Johns, NL, Canada.
mhahmed@mun.ca

Tamer Khattab
Faculty of Engineering
Qatar University,
Doha, Qatar.
tkhattab@ieee.org

***Abstract*—Proactive caching is a promising technique used to minimize peak traffic rates by storing popular data, in advance, at different nodes in the network. We study a cellular network with one base station (BS) communicating with multiple mobile units (MUs). The BS has a number of cached files to be delivered to the MUs upon demand, and the popularities of these files are changing over time. We show that proactively and constantly updating the MU finite caches and jointly encoding the delivery of different demanded files to the MUs over different time slots minimize the delivery sum rate. We propose two different schemes for a two different scenarios, where the file popularities over time can be either arbitrary increasing or decreasing for the first scheme and decreases with demand for the second scheme. Numerical results show the benefits of the proposed schemes, over conventional caching schemes, in terms of reducing the delivery sum rate.**

I. Introduction

Cellular network traffic has shifted over the past decade from mainly locally generated instantaneous traffic (voice calls) to centrally generated delay-tolerant bulks of traffic (data communication) [1]. This brought a need for change in the information theoretic analysis of communication networks. Contrary to voice calls, mostly the delay between time of data generation and time of data demand at the receivers is large. In this context, the data can be stored midway in different nodes along with the transmitter and receivers. Moreover, the shift from speech to data traffic contradicts the classical assumption of network analysis that the messages are generally independent. In a sense, the information paradigm is more complex where the demand for a certain piece of information generally probabilistic not deterministic and the broadcast messages is more dominant than the unicast messages. Proactive caching is an efficient technique to reduce the peak traffic rate and the delivery sum rate. This is achieved by storing parts of the popular content, at various nodes in the networks, before being requested by the users. In practical sense, proactive caching can minimize the total cost of transmission, as transmitters can optimize the time of caching to be in less congested times.

Wireless networks with caches were studied extensively in recent research [2]–[8]. In [2], a novel two phase communication model that mirrors the probabilistic and broadcast message characteristics was studied. A number of independently generated messages, each corresponding to one piece of content, is available during the first transmission phase, i.e. the content placement phase. During the second transmission phase, i.e., the content delivery phase, each receiver has a deterministic demand for one of the message. The main difference compared to the traditional communications is that the transmitter and receiver need to optimize the channel use for both phases jointly, where the demand is probabilistic for the first phase. The authors in [2] formally introduced the communication over a cache-aided interference channel and derive its degrees of freedom in terms of the cache size. An information-theoretic framework for the analysis of cache-aided communication was introduced in [3] in the context of broadcast channels. It is shown in [3] that the availability of caches in a broadcast setting provides a coded multi-casting gain. While the aforementioned works considered a one time slot delivery phase, other works studied a more general multiple time slot delivery phase [4]–[8]. In [4], the content placement is optimized while the transmitter being oblivious of the demands statistics. In [5], a cache-aided small-cell system is considered, and the cache placement is formulated as maximizing the weighted-sum of the probability of local delivery. The authors in [6] studied the load-balancing benefits of caching in ad-hoc wireless networks. In [7], content placement is optimized for fixed capacities small cells. In [8], caching for peer to peer systems was studied, the authors modeled the traffic in peer to peer systems and developed a modified complementing peer to peer caching algorithm.

In this paper, we assume that the delivery phase is composed of multiple time slots, where each user requests a file delivery in a different time slot. As such, joint broadcast coding, e.g., [3], cannot be used. While the works available in literature delivers the requested information via independent messages in the delivery phase, we show that we can still get the benefit of joint broadcast coding if the placement and delivery phases are carefully designed to accommodate for the delivery over multiple time slots. This is achieved by exploiting the broadcast nature of the channel where a receiver can make use of the information sent to other receivers to update its own cache. Moreover, we show that constantly updating the caches instead of using one time slot placement phase further minimize the delivery rate. In particular, we consider two demand scenarios, and propose two caching schemes that

aim to minimize the expected delivery sum rate. In the first demand scenario, we focus on the case of two mobile units (MUs) with changing file popularities, i.e. with a increasing or decreasing probability that a give file is requested. The MUs store parts of the files cached at the BS in their caches before making any demands. Missing parts of the requested files will be transmitted by the BS at the time of demand. The proposed scheme enables joint coding (i.e., encode the missing information of a certain MU and update the caches of other MUs) even though the files are delivered at different time slots, and hence, reduces the expected delivery sum rate. In the second demand scenario, we study the K MUs case with decreasing file popularities. The proposed scheme jointly encodes the delivery of files over different time slots and shows a reduction in the expected delivery sum rate. Numerical results are provided to show the merits of the proposed schemes, over conventional caching schemes, in terms of reducing the delivery sum rate.

The remainder of the paper is organized as follows, the system model is described in section II, the first demand scenario is investigated in section III, while the second demand scenario is studied in section IV. The numerical results are presented in section V and we conclude in section VI.

II. System model

We consider a broadcast channel, where a base station (BS) communicates with K mobile units (MUs). The BS and MUs are assumed to be equipped with one antenna each and have limited size caches. The BS has a database of K cached files $\mathcal{F} = \{F^1, F^2, ..., F^K\}$, each of size N bits; while each MU has an N bit cache that equals the size of one file. A block diagram describing the system model is shown in Fig. 1.

The transmission from the BS to the MUs occurs over two phases, the placement phase and the delivery phase. During the first phase, i.e., the placement phase, the caches of the MUs are filled with information that is a function of the files $\mathcal{F}$ stored at the BS up to the caches size N. During the second phase, i.e., delivery phase, a MU requests one of the files $\{F^i; i \in \{1, 2, ..., K\}\}$ and during this phase the BS sends any missing information of the file that can not be extracted from the user local cache. The MUs files demands are assumed to arrive at the beginning of different time slots, and the BS satisfies each request during its time slot at the placement phase. The files requested by the MUs are not known at the placement phase, but their popularity (the probability that a MU requests a file) at any time instant is known a priori to the BS.

We consider two different file demands scenarios. In the first scenario, we assume that the file popularity is changing over time (i.e., it can be either increasing of decreasing). While for the second scenario, we assume the file popularity decreases after it is demanded, that is to say the probability that a file is demanded m times is less than the probability it is demanded $m-1$ times, and all files are assumed to be equally popular at the onset of the delivery phase.

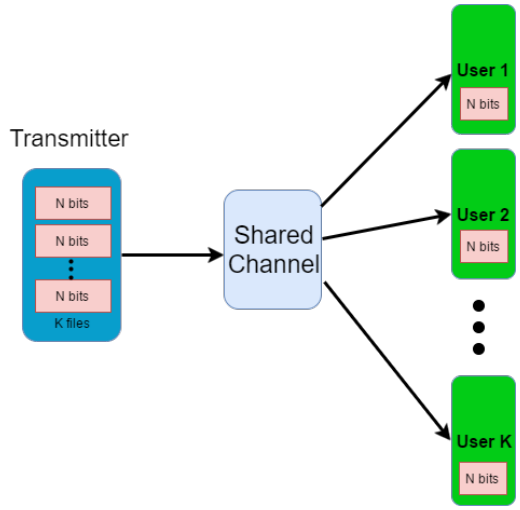


Fig. 1: Cache aided cellular network

Our objective is to design the information transfer through the placement and delivery phases to minimize the amount of information sent in the delivery phase. We propose schemes, for the aforementioned file demands scenarios, that jointly encode the missing information of a given MU and the update of the other MUs caches.

III. The First Demand Scenario

The caching problem has been extensively studied for files popularities that do not change over time [2], [3], [5]–[7]. As a result, caching the most popular files at the placement phase was found to br optimal in terms of the delivery phase rate. Antithetically, caching the most popular files at the placement phase is not optimal when the files popularities change over time. For instance, in a more practical situation the average files popularities in a certain geographical area covered by a certain BS or in a certain sport event changes and/or fades with time. Or, the average files popularities of a new movie released on Netflix may fade over time, as a MU will be less likely to re-watch the movie after the first time. That said, in the majority of practical applications, the assumption of the non-changing files popularities will lead to a sub-optimal solution of the sum rate of the delivery phase.

In this section, we study the first demand scenario for the two users case. For the ease of notations, the two files available for demand at the BS are denoted $\{A, B\}$ instead of $\{F^1, F^2\}$. Let P_i^A and P_i^B be the probability of demand for file A and B at the ith time slot, respectively. Assume without loss of generality that file A is more popular at the first time slot, while file B becomes more popular in the second one.

A. Caching whole files

If file A is cached at both MUs at the placement phase; and file A is requested during the first time slot, then the first time slot delivery rate will be zero. On the other hand, if file B is requested during the first time slot, then the first time slot delivery rate will be $P_1^B N$. Hence, the expected delivery rate during the first time slot is $P_1^B N$. Similarly, the

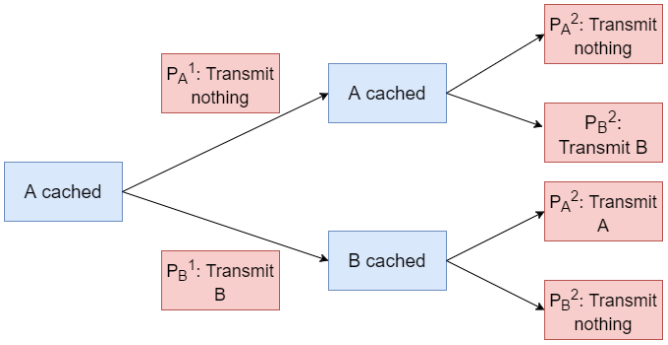


Fig. 2: Delivery probability tree for two user channel

second time slot expected delivery rate depends on the cached content. If file A is demanded in the first time slot, the second cache would contain file A. On the other hand, if file B was demanded in the first time slot by the first MU, then the second MU cache can be updated to store file B. For the remainder of this section, all rates are normalized(with respect to N). Fig. 2 shows all the demand possibilities during the two time slots and the associated delivery. Consequently, the expected sum rate of the two delivery time slots if file A is cached at the placement phase can be written as

$$\begin{aligned} R_A &= P_1^A P_2^B + P_1^B P_2^A + P_1^B && (1)\\ &= P_1^A(1-P_2^A) + (1-P_1^A)P_2^A + (1-P_1^A) && (2)\\ &= 1 + P_2^A - 2P_1^A P_2^A, && (3) \end{aligned}$$

while if file B is cached at the placement phase. The expected delivery sum rate can be expressed as

$$\begin{aligned} R_B &= P_1^A P_2^A + P_1^B P_2^A + P_1^A && (4)\\ &= P_1^A P_2^A + (1-P_1^A)P_2^A + P_1^A && (5)\\ &= P_1^A + P_2^A. && (6) \end{aligned}$$

The delivery phase rate can be minimized by choosing the file to be cached at the placement phase depending on the file popularities (P_1^A, P_2^A). Hence, the minimum expected delivery sum rate if one file is cached at the placement phase can be formulated as

$$R^*_{\text{whole}} = \min\{1 + P_2^A - 2P_1^A P_2^A, P_1^A + P_2^A\}. \quad (7)$$

B. Caching partial files with joint coding

While the previous scheme is restricted to caching the whole files. We will show that if files are split and MUs cache parts of each file, the expected delivery sum rate is further reduced.

1) Placement Phase: Let S_A and S_B be the normalized sizes (with respect to N) of parts of files A and B, respectively, that are cached at the placement phase at each MU, i.e., $S_A + S_B = 1$ (as shown in Fig. 3). Our target here is to find the optimal cache at the placement phase (in terms of S_A and S_B) to minimize the expected delivery sum rate.

Let $S_A > S_B$ at the placement phase and $\{\bar{A}_1, \bar{A}_2\}$ are two non-overlapping equally sized parts of file A, i.e., $|\bar{A}_1| = |\bar{A}_2| = \frac{1}{2}$, where $|X|$ denotes the cardinality of file X. Also assume that the BS caches parts of file A, i.e., $\{A_1, A_2\}$, both

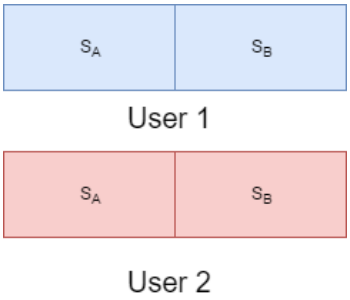


Fig. 3: Users cache at the placement phase

of size S_A at the first and second MUs, respectively, such that

$$\bar{A}_i \subseteq A_i; \quad i = 1, 2, \quad (8)$$

where $A \subseteq B$ means that B has all the information in A. Equation (8) implies that file A is completely cached into the caches of the two MUs. Similarly, $\{\bar{B}_1, \bar{B}_2\}$ are two non-overlapping equally sized parts of file B, i.e., $|\bar{B}_1| = |\bar{B}_2| = \frac{1}{2}$, and the BS caches parts of file B, i.e., $\{B_1, B_2\}$, both of size S_B at the first and second MUs, respectively, such that

$$B_i \subseteq B_i^*; \quad i = 1, 2, \quad (9)$$

Equation (9) implies that file B is partially cached into the caches of the two MUs.

2) Delivery Phase: If the first MU demands file A at the first delivery time slot, then the BS transmits

$$C = (A - A_1) \oplus B_1, \quad (10)$$

and the first MU XOR C with B_1 to extract $A - A_1$, where $\oplus$ represent bit by bit XOR, and $A - B$ represent the information in A that is not B. The second MU XOR C with $A - A_1$ (which is a part of A_2) to extract B_1, and then update its cache by replacing a part of A_2 with B_1 as the popularity of file B increases in the second time slot. As a result, the updated cache of the second MU has B_1 and B_2 of size $2S_B$ and a part of A of size $(1-2S_B)$. At the second time slot, the BS transmits a part of size $2S_B$ if file A is demanded by the second MU or a part of size $(1-2S_B)$ if file B is requested.

On the other hand, if file B is demanded by the first MU at the first delivery time slot, then the BS sends

$$C = \{B_2 \oplus B_1, B - \{B_1, B_2\}\}. \quad (11)$$

As such, the first MU extracts $B - B_1$, while the second MU updates its cache to include the whole file B that becomes more popular for the coming second delivery time slot and caching it is instantly optimal. At the second time slot, the BS transmits nothing if file B is requested by the second MU, while it transmits the whole file A if it was requested by the second MU. The delivery phase sum rate for this scheme is

$$\begin{aligned} R_{AB} &= P_1^A S_B + P_1^B S_A + P_1^B P_2^A + 2(P_1^A P_2^A)S_B \\ &\quad + P_1^A P_2^B(1 - 2S_B). \quad (12) \end{aligned}$$

Given that $P_i^B = 1 - P_i^A$ and $S_A = 1 - S_B$, then

$$R_{AB} = 1 + P_2^A - 2P_1^A P_2^A + (4P_1^A P_2^A - 1)S_B. \quad (13)$$

Hence, the minimum expected delivery sum rate if parts of files A and B are cached at the placement phase and $S_A > S_B$ can be formulated as

$$\begin{aligned} R^*_{AB} &= \min_{S_B} 1 + P_2^A - 2P_1^A P_2^A + (4P_1^A P_2^A - 1)S_B \\ &\quad \text{subject to} \quad S_B \leq \frac{1}{2}. \end{aligned} \tag{14}$$

The solution of the expected delivery sum rate minimization problem in (14) depends on the sign of $(4P_1^A P_2^A - 1)$, as the problem represents a one-dimensional linear optimization problem. If $4P_1^A P_2^A \geq 1$, then the term $(4P_1^A P_2^A - 1)S_B$ is positive and the minimizing value of S_B is 0, i.e., $S_B^* = 0$ and the resulting minimum delivery sum rate is given as

$$R^*_{AB} = 1 + P_2^A - 2P_1^A P_2^A, \quad \text{if} \quad 4P_1^A P_2^A \geq 1, \tag{15}$$

which is intuitively equals the value of its counterpart in (3). On the other hand, if $4P_1^A P_2^A < 1$, then term $(4P_1^A P_2^A - 1)S_B$ is negative and the minimizing value of S_B is $\frac{1}{2}$, i.e., $S_B^* = \frac{1}{2}$, and the resulting minimum delivery sum rate is given as

$$R^*_{AB} = \frac{1}{2} + P_2^A, \quad \text{if} \quad 4P_1^A P_2^A < 1. \tag{16}$$

Similarly if $S_A \leq S_B$ at the placement phase, i.e file B is fully cached while file A is partially cached at the combined cache of the two MUs, the delivery phase sum rate would be

$$R_{AB} = P_1^A S_B + P_1^B S_A + P_1^A P_2^A + P_1^B P_2^A, \tag{17}$$

and given that $P_B^i = 1 - P_A^i$ and $S_A = 1 - S_B$, then,

$$R_{AB} = (2P_1^A - 1)S_B - P_1^A + P_2^A + 1. \tag{18}$$

Hence, the minimum expected delivery sum rate if parts of files A and B are cached at the placement phase and $S_A \leq S_B$ can be formulated as

$$\begin{aligned} R^*_{AB} &= \min_{S_B} (2P_1^A - 1)S_B - P_1^A + P_2^A + 1 \\ &\quad \text{subject to} \quad S_B \geq \frac{1}{2}. \end{aligned} \tag{19}$$

Since file A is more popular at the first delivery time slot, then $P_1^A > \frac{1}{2}$ and $(2P_1^A - 1)$ is positive. The minimizing value for S_B is $\frac{1}{2}$, i.e. $S_B^* = \frac{1}{2}$. Accordingly, the minimum delivery sum rate is

$$R^*_{AB} = \frac{1}{2} + P_2^A. \tag{20}$$

By comparing (3), (6), (16), and (20), one can see that caching file B which is the less popular at the first delivery time slot (in (6)) always yields higher delivery sum rate compared to storing file A (in (3)) or storing half of each file (in (16) and (20)). Moreover, caching the whole file A is optimal if $P_1^A P_2^A > 1$; while caching half of each file is optimal if $P_1^A P_2^A < 1$. Finally, the minimum delivery sum rate can be expressed as

$$R^* = P_2^A + \min\left\{\frac{1}{2}, 1 - 2P_1^A P_2^A\right\}. \tag{21}$$

IV. The Second Demand Scenario

The conventional caching schemes in the literature split the cache of each MU equally between the K files. This results in a delivery rate of $\frac{K-1}{K}N$ per MU, and hence, the delivery sum rate for conventional caching would be

$$R_{\text{con}} = (K-1)N. \tag{22}$$

In this section, we propose a caching scheme that is able to further reduce the delivery sum rate of the MUs.

3) The Placement Phase: The BS splits the lth file into a number of $K(K-1)$ subfiles $\{F_{ij}^l; l = 1, 2, ..., K; i = 1, 2, ..., K; j = 1, 2, ..., K-1\}$. In the placement phase, MU m stores subfiles $\{F_{mj}^l; l = 1, 2, ..., K; j = 1, 2, ..., K-1\}$.

4) The Delivery Phase: As the delivery phase is assumed to happen over K time slots, at the ith time slot of the delivery phase, the BS delivers the file demanded by the ith MU while updating the caches of the remaining $K - i$ MUs. For example, assume without the loss of generality, that the first MU requests file F^1 at the first time slot. The BS has to deliver the remaining $(K-1)$ subfiles $\{F_{ij}^1, i = 2, 3, ..., K;\ j = 1, 2, ..., K-1\}$ to the first MU, while at the same time updates the caches of the remaining $K-1$ MUs. Accordingly, the BS transmits

$$\begin{aligned} &\{F_{21}^1, F_{22}^1, ..., F_{2K-1}^1, F_{31}^1, F_{32}^1, ..., F_{3K-1}^1, ..., \\ &F_{K1}^1, F_{K2}^1, ..., F_{KK-1}^1\} \oplus \{F_{11}^2, F_{11}^3, ..., F_{11}^K \\ &F_{12}^2, F_{12}^3, ..., F_{12}^K, ..., F_{1K}^2, F_{1K}^3, ..., F_{1K}^K\}. \end{aligned}$$

Since The second part of the XOR $\{F_{11}^2, F_{11}^3, ..., F_{11}^K, F_{12}^2, F_{12}^3, ..., F_{12}^K, ..., F_{1K}^2, F_{1K}^3, ..., F_{1K}^K\}$ is already cached at the first MU, it will be able to decode the first part which contains the missing parts of F^1. On the other hand, MU $l; l = 2, ..., K$ will be able to replace the subfiles of file F^1 stored at its cache with subfiles of the other files $F^l; l = 2, 3, ..., K$. In the previous process each MU other than the first MU will update $\frac{1}{K}$ of its cache. After the update the cache will be equally shared between $K-1$ files totally eliminating file F^1, which reduces the delivery rate for the next delivery time slots.

In a similar fashion, in the next time slots the cache of the remaining MUs is updated by eliminating the demanded file and replacing it with parts of the remaining files. At the t time slot time slot, an MU would either demand a partially cached file with probability P_t^{new} or has a repeated demand for an eliminated file with probability P_t^{old}. At each time slot, the cache of the remaining MU is equally split between $K-l$ files, where l is the number of files that was already demanded. Let L be the expected number of files to be demanded once, the expected delivery sum rate for decreasing file popularities would be

$$R_{\rm d}(L) = \sum_{l=0}^{L-1} \frac{K-l-1}{K-l} N + (K-L)N \quad (23)$$

$$= \left(K - \sum_{l=0}^{L-1} \frac{1}{K-l} + (K-L) \right) N \quad (24)$$

$$= \left(2K - L - \sum_{C=K-L+1}^{K} \frac{1}{C} \right) N \quad (25)$$

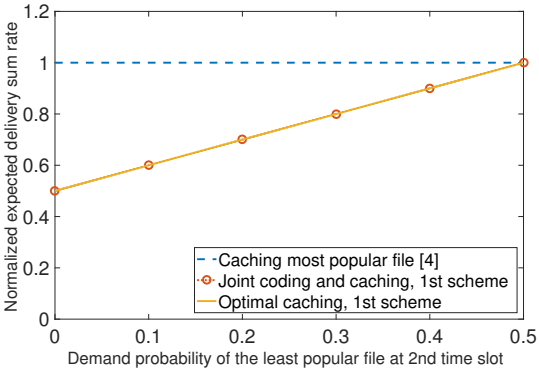
$$= \left(2K - L - (\psi(K+1) - \psi(K-L+1)) \right) N \quad (26)$$

where ψ is the digamma function, and the expected delivery sum rate if all files are demanded only once is

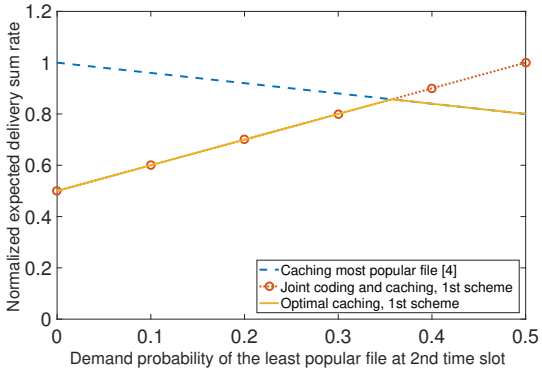
$$R_{\rm d}(K) = \left(K - (\psi(K+1) + \gamma) \right) N, \quad (27)$$

where γ is the Euler-Mascheroni constant. As discussed in the beginning of this section, the expected delivery sum rate without using our proposed scheme is $(K-1)N$; hence, the coding gain $R_{\rm gain}$ of our proposed scheme if all files are demanded only once is given as

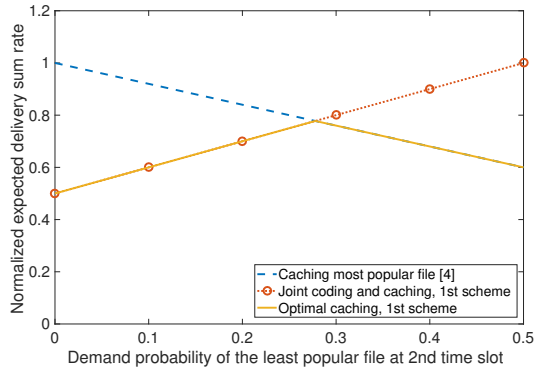
$$R_{\rm gain} = R - (K-1)N$$
$$= (\gamma + \psi(K+1) - 1)N. \quad (28)$$



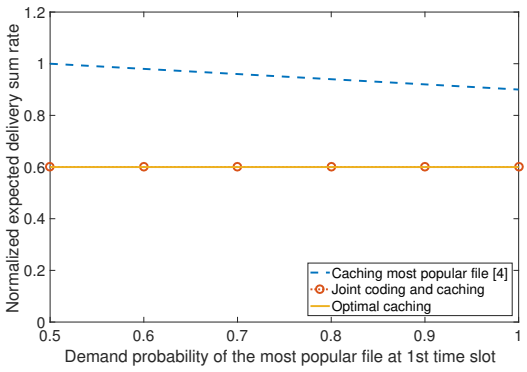
(a) $P_1^A = 0.5$.



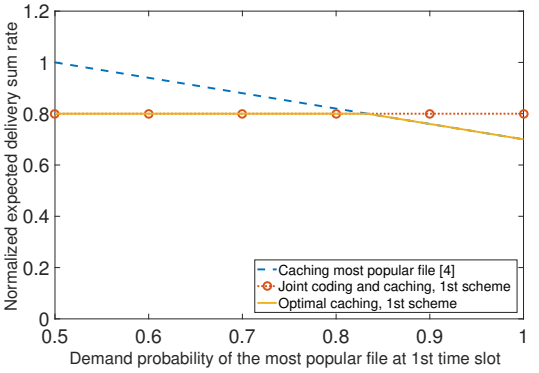
(b) $P_1^A = 0.7$.



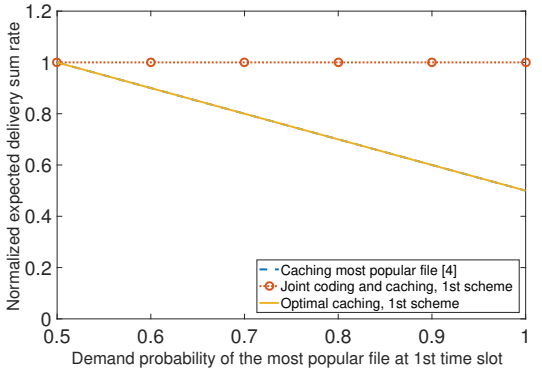
(c) $P_1^A = 0.9$.



(d) $P_2^A = 0.1$.



(e) $P_2^A = 0.3$.



(f) $P_2^A = 0.5$.

Fig. 4: Normalized expected delivery sum rate of the first scenario for different values of the file popularities.

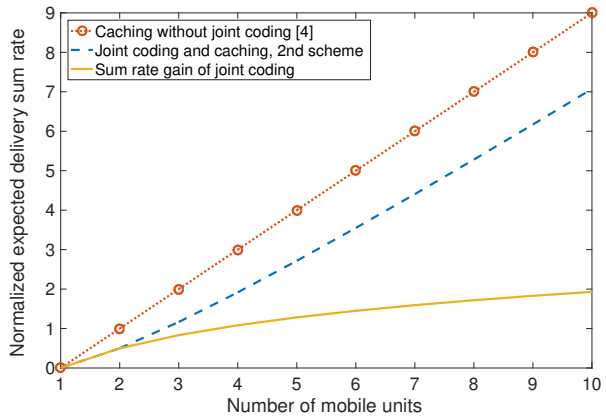


Fig. 5: Normalized expected delivery sum rate of the second scenario.

V. Numerical Results

In this section, we provide numerical results that show the merits, in terms of the expected delivery sum rate, of the proposed schemes. Fig. 4 compares the expected delivery sum rate for the first demand scenario for different values of the file popularities. Three schemes are presented in Fig. 4: "Caching most popular file [4]" represents caching of file A only at the placement phase as in [4] or as in (3), " Joint coding and caching, 1st scheme" denotes caching parts of both files A and B and jointly delivering both files as in (20), and "Optimal caching, 1st scheme" is for alternating between the previous schemes depending on the file popularities as in (21).

Fig. 4 shows the performance of our two users scheme and compares it to the (caching most popular file) scheme. Each of fig. 4a, 4b, 4c shows the performance of a certain value for P_1^A and different values for P_2^A, while fig. 4a, 4b, 4c shows the performance of a certain value for P_2^A and different values for P_1^A. As can be seen in Fig. 4a, joint coding and caching is always optimal regardless the value of P_2^A. Similarly, in Fig. 4d joint coding and caching is always optimal regardless the value of P_1^A. On the other hand, for larger values of P_1^A in Fig. 4b and Fig. 4c, the optimality is conditioned on the value of P_2^A. Similarly, for larger values of P_2^A in Fig. 4e the optimality is conditioned on the value of P_2^A. Finally in Fig. 4f, caching the most popular file (for $P_2^A = 0.5$) is always optimal regardless the value of P_1^A. The figures shows that our scheme can minimize the transmission rate used in proactive caching for a big portion of the file popularity values where the gain can reach .5 of the rate used by (caching the most popular file) scheme.

Fig. 5 compares the expected delivery sum rate of the second demand scenario for the case of joint coding and caching in (23) to caching without joint coding as in [4] or in (22). As can be seen, joint coding and caching is superior in terms of minimizing the expected delivery sum rate. Fig. 5 additionally reveals that the sum rate gain R_{gain} grows with increasing the number of MUs.

VI. Conclusion

In this paper, we studied proactive caching for a cellular network with one BS and multiple MUs. We proved that proactively updating the local finite caches and jointly encoding the files delivery to the MUs over different time slots minimize the delivery sum rate. In particular, for the first scenario where the file popularities are increasing or decreasing over time, storing the less popular file is always inferior whatever changes happen to the popularities over time.

Further, using joint encoding is superior for some values of file popularities, while storing the most popular files at the beginning of delivery phase is superior for the other values. On the other hand, for the second scenario, replacing the less popular files with their more popular counterparts minimizes the expected delivery sum rate.

References

[1] Cisco Visual Networking Index, "The zettabyte era–trends and analysis," *Cisco white paper*, 2013.

[2] M. A. Maddah-Ali and U. Niesen, "Cache-aided interference channels," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, 2015, pp. 809–813.

[3] ——, "Fundamental limits of caching," *IEEE Transactions on information theory*, vol. 60, no. 5, pp. 2856–2867, May 2014.

[4] P. Blasco and D. Gunduz, "Learning-based optimization of cache content in a small cell base station," in *Proc. IEEE International Conference on Communications (ICC)*, 2014, pp. 1897–1903.

[5] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless video content delivery through distributed caching helpers," in *Proc. IEEE International Conference on Computer Communications (INFOCOM)*, 2012, pp. 1107–1115.

[6] U. Niesen, D. Shah, and G. W. Wornell, "Caching in wireless networks," *IEEE Transactions on information theory*, vol. 58, no. 10, pp. 6524–6540, Oct. 2012.

[7] K. Poularakis, G. Iosifidis, and L. Tassiulas, "Approximation algorithms for mobile data caching in small cell networks," *IEEE Transactions on communications*, vol. 62, no. 10, pp. 3665–3677, Oct. 2014.

[8] M. Hefeeda and O. Saleh, "Traffic modeling and proportional partial caching for peer-to-peer systems," *IEEE/ACM Transactions on networking*, vol. 16, no. 6, pp. 1447–1460, Dec. 2008.